

INTERNET ENDPOINT SYSTEM

BACKGROUND OF THE INVENTION

- [01] The present invention relates generally to a system for reducing packet congestion for multiple devices that need to communicate over a network, such as the Internet. The subject matter of this application is related to my previously-filed U.S. application number 10/697,103, entitled "Endpoint Packet Scheduling System," filed on October 31, 2003.
- [02] Ethernet and packet-switched Internet Protocol (IP) networks are systems for transmitting data between different points. These systems are known as "contention-based" systems. That is, all transmitters contend for network resources. All transmitters may transmit simultaneously. If they do, then network resources may be oversubscribed. When this happens, data may be delayed or lost, resulting in network impairment.
- [03] As shown in FIG. 1, a conventional network comprises a plurality of Local Area Network (LAN) endpoints, such as computers connected to an Ethernet LAN. The endpoints may comprise, for example, one or more IP-enabled telephones; video cameras; televisions; desktop computers; printers; or any other device that communicates over the LAN. The data may comprise real-time or quasi-realtime data, such as video frames, audio frames (voice-over-IP connections), and other types of data. The endpoints are coupled to one or more LAN switches 105, which connect the LAN to a Wide Area Network (WAN) through a router 106. When multiple endpoint devices transmit packets destined for the WAN, congestion can occur on the uplink 107 and 108.
- [04] When endpoint 101 sends packets destined for the WAN, the packets are sent through LAN switch 105, which also handles packets from other LAN endpoints. If too many packets are simultaneously transmitted by the other endpoints to LAN switch 105, LAN switch 105 may have a queue overflow, causing packets to be lost. Similarly,

WAN router 106 may have a queue overflow, causing packets to be lost at that point. The net result is that congestion due to contention among local transmitters causes data to be lost.

- [05] The word "packets" will be used to refer to datagrams in a LAN or Wide Area Network (WAN) environment. In a LAN environment, packets are sometimes called "frames." In a packet-switched WAN environment, packet-switching devices are normally referred to as "routers." In some systems, the LAN switching and WAN routing functions can be combined into a single device.

- [06] FIG. 2 illustrates the nature of the problem of dropped packets, which can occur in a LAN environment as well as a WAN environment. During periods where multiple endpoints are simultaneously transmitting packets on the network, the LAN switch 105 and/or WAN router 106 may become overloaded, such that some packets are discarded. This is typically caused by an internal queue in the devices becoming full and thus becoming unable to accept new packets until the outgoing packets have been removed from the queue. This creates a problem in that transmitting endpoints cannot be guaranteed that their packets will arrive, necessitating other solutions such as the use of guaranteed-delivery protocols such as Transmission Control Protocol (TCP). Such solutions may be inappropriate for broadcast video, voice over IP or other realtime applications, which cannot wait for retransmission of packets.

- [07] FIG. 2 depicts the situation where multiple packet flows create congestion and loss within an Ethernet switch. FIG. 2 shows four different flows on four separate Ethernet ports, Input A through Input D (elements 201 through 204), the resulting output flow, Output E (element 205) and the discarded frames 206. Note that Inputs A-D (elements 201 through 204) have bursts of packets that overlap one another time-wise. The utilization of each input 201-204 is within limits. That is, each input is not saturated or oversubscribed. The total utilization of each input 201-204 is also not too much data in aggregate to oversubscribe the Output E (element 205). However, because of the coincidental timing of the packet bursts on Input A through Input D, the Ethernet switch is oversubscribed. Thus packets will be lost when the Ethernet

switch queue overflows lost packets, shown as discarded frames 206. Note that frames are not immediately discarded from the combined input frames. The switch uses a queue which is used to hold frames until a later transmission can occur. When concurrent bursts arrive, the queue can fill and cause frames to be discarded as illustrated in element 206.

SUMMARY OF THE INVENTION

- [08] The invention provides a method for transmitting packets in a network by eliminating contention among local transmitters in the network. A device comprises a CPU and other components that accept input from various transmitting data sources. The device processes signals from the various transmitting sources and converts these signals to packets, before they are output via a LAN and/or WAN interface, thus avoiding contention on the LAN, since only one device produces the packets from many signal sources. This device may initiate and receive connections via the WAN and begin transmitting over the Ethernet to the WAN at agreed-upon rates, without contention.

BRIEF DESCRIPTION OF THE DRAWINGS

- [09] FIG. 1 shows the problem of multiple packet sources creating an overflow condition at a packet switch, leading to packet loss.
- [10] FIG. 2 shows how network congestion can cause packet loss where multiple sets of endpoints share a common network resource under bursty conditions.
- [11] FIG. 3 shows one embodiment of a device for carrying out various principles of the invention.
- [12] FIG. 4 shows one example of using a device 301 to reduce contention for resources when a plurality of transmitting sources 401 through 404 are transmitting and/or receiving packets.

- [13] FIG. 5 shows a system employing multiple devices coupled together and synchronized to avoid overloading a LAN switch.
- [14] FIG. 6 shows one method for coordinating the transmission of data from the devices shown in FIG. 5.
- [15] FIG. 7 shows one possible transmission map for use in the method of FIG. 6.

DETAILED DESCRIPTION OF THE INVENTION

- [16] FIG. 3 shows one possible device for carrying out various principles of the present invention. The device 301 comprises a backplane bus 307 and a CPU 302 that coordinates the scheduling and other activities of the device. CPU 302 may comprise, for example, an IBM 440GX processor. The CPU includes or is coupled to a local area network, such as an Ethernet via an Ethernet interface that outputs Internet Protocol packets encapsulated in Ethernet frames, as is conventional. The Ethernet interface may for example comprise a gigabit Ethernet. Other network interfaces are possible, such as a telecommunication interface, or any type of packet interface. The packet network interface may be coupled to a WAN router 313 for transmission and reception over a Wide Area Network (WAN), such as the Internet. In another embodiment, device 301 may be directly coupled to WAN network router 313 rather than being connected through device 314. In yet another embodiment, device 301 may be directly coupled to the WAN without requiring external devices. In general, device 301 may be coupled through any packet network interface according to the inventive principles.
- [17] Device 301 also includes a plurality of modules 303 through 312 that are coupled to a backplane bus 307. The backplane bus may comprise for example a 32-bit wide bus operating at 33 MHz. The modules coupled to the bus may include one or more of the following. A timing system 303 provides clock and interval signals to and from other modules as shown in more detail in FIG. 5. In one embodiment, timing system 303 works as follows. The timing system will receive a reference timing signal from an

external source. In many cases this timing source can be a telecom timing reference or a GPS clock. The timing reference enters the timing system 303 as a clock signal. This clock can be a 1 MHz, 5 MHz or 10 MHz signal. It can also be a T1 or E1 clock. The timing system 303 employs a set of clock dividers and phase-locked multipliers to generate the clocks needed by each of the modules 304-312 in the system of FIG. 3 301. Each module can also derive its own timing clock individually by referencing the clock that is provided by the timing system 303 through the backplane 307. The timing system 303 can also be commanded, by the CPU 302, through the backplane 307 to provide timing interrupts at precise intervals, to the CPU 302. These interrupts can be used to initiate transmission of packets at precise intervals to the Ethernet Network. The CPU 302 can also use the precise interrupts, to control its access of data from the various modules 304-312. In this way, the CPU can always have sufficient data from each module to create a packet, at the time the CPU has set to transmit the packet.

- [18] A parallel data card 304 accepts parallel data from one or more modules and presents the data to CPU 302 via backplane bus 307. Examples of parallel data include parallel ports on page printers, scientific instruments and memory loading and CPU emulation devices. Asynchronous data card 305 accepts asynchronous data from one or more devices and presents the data to CPU 302 via backplane bus 307. Examples of such data include serial device controllers, communications interfaces to display terminals, keyboards, teletype devices, IR remote control devices, etc. As with the other devices described below, in one embodiment data card 305 can also transmit asynchronous data to one or more devices from the backplane bus 307.

- [19] Ethernet data card 306 accepts Ethernet data frames (e.g., ad hoc computer data) and presents the data to CPU 302 via backplane bus 307. Video data card 308 accepts video analog or digital signals from one or more video devices (e.g., video cameras or the like), converts this into digital data, and may compress the digital data before presenting the data to CPU 302 via backplane bus 307. Stereo card 309 accepts stereo analog or digital signals from one or more sources (e.g., CD players, radios, or the like), and may convert the stereo signals to digital data and presents the data to CPU

302 via backplane bus 307. Analog card 310 converts analog data (e.g., voice, measurements, or other analog data) into digital form and presents the data to CPU 302 via backplane bus 307. T1/E1 card 311 receives data in T1/E1 format from one or more sources and presents it to CPU 302 via backplane bus 307. Synchronous data card 312 accepts synchronous data from various sources and presents it to CPU 302 via backplane 307. The various modules shown in FIG. 3 are intended to be illustrative only. More or fewer modules may be included, and other types of data beyond those illustrated may be included.

- [20] In one embodiment, CPU 302 organizes the work that is done by the modules connected to backplane bus 307. CPU 302 detects the presence of data at one or more of the modules; retrieves the data in a format suitable for the particular module; reformats the data into Ethernet packet frames; and transmits each packet via Ethernet to an Ethernet LAN switch 314 to a WAN network router 313, or via Ethernet directly to a WAN network router 313. The data can be transmitted over the bus using DMA techniques; interrupts; or other methods. Because only one CPU handles the processing, there is no contention for resources. CPU 302 is also responsible for executing any call establishment protocols. These protocols can include MGCP, H.323, Session Initiation Protocol (SIP), etc. These protocols are used to perform call set up and disconnect.
- [21] FIG. 4 shows one example of using a device 301 to reduce contention for resources when a plurality of transmitting sources 401 through 404 are transmitting and/or receiving information. Device 301 is the same device 301 of FIG. 3, and is coupled to a plurality of transmitting sources 401 through 404, such as telephones, and video cameras that transmit video frames. Instead of coupling each transmitting source directly to an Ethernet, which might cause packet loss in an Ethernet LAN switch, each device is coupled to device 301, which reduces congestion by placing packets from each module on the Ethernet using a single CPU and data bus. Realtime data or near-realtime data can thus be handled without losing packets due to LAN contention. As connections are established with devices across the WAN, data flows through the device 301 to and from the devices 401 to 404.

- [22] FIG. 5 shows a system employing multiple devices coupled together. When the number of realtime I/O interfaces exceeds what an individual endpoint system 501 can support, then multiple endpoint systems 502-504 can be connected so that they work cooperatively. They can be synchronized to avoid creating congestion or packet loss in a LAN switch. As shown in FIG. 5, each device 501 through 504 corresponds to device 301 of FIG. 3. Each device includes a clock/frame interface corresponding to timing circuit 303 of FIG. 3. One of the devices 501 is designated as the master, and the others are designated as slaves. As with the devices in FIG. 3, each device 501 through 504 accepts inputs from and generates outputs to multiple devices of various types. Each device generates an output to (and receives input via) an Ethernet link to LAN switch 505, which in turn is linked to a WAN router. In order to avoid congestion in LAN switch 505, each device 501 through 504 coordinates its schedule such that packets are not transmitted at the same time to LAN switch 505. Any of various methods can be used to accomplish this, such as assigning time slots for each device to transmit to LAN switch 505 and using the clock/frame interfaces to coordinate this timing.
- [23] FIG. 6 shows one method of coordinating a timing schedule among multiple devices to avoid congestion in LAN switch 505. Beginning in step 601, when one of the slave devices in FIG. 5 intends to send data going to or from the I/O devices that it serves, it queries the master device 501 for a transmission map indicating time slots during which the master and/or other slaves have already allocated packet transmissions. A delivery schedule comprising an arbitrary time period (e.g., one second) can be decomposed into frames, subframes and so forth, and each device can determine based on the bandwidth required to support a particular connection how many time slots are needed to transmit data to and from LAN switch 505. (Further details of one possible scheme are described in my previously filed application serial number 10/697,103, entitled "Endpoint Packet Scheduling System," filed on October 31, 2003, which is incorporated by reference herein.)
- [24] The master device responds in step 602 with a transmission map showing what time slots are available for transmission to LAN switch 505 or, alternatively, what time

slots have already been allocated. In step 603, the slave sends a proposed transmission schedule indicating what time slots it proposes to use in order to carry data sufficient to support the various devices it is serving. In step 604, the master device agrees to the proposed schedule or proposes an alternate schedule. FIG. 7 shows one possible implementation of a transmission map, which is essentially a bitmap with each bit corresponding to a time slot.

- [25] In step 605, the slave device uses the assigned time slots to schedule packets for transmission over the Ethernet to and from LAN switch 505. In step 606, the slave ends transmission when the connection is no longer needed. In step 607, the slave signals the master to indicate that no further transmission will be required. In step 608, the master removes the slave from the transmission map, freeing up the time slots for other slaves.

- [26] As an alternative to a proposed transmission map, each slave device could request a bandwidth and receive from the master a proposed schedule taking into account the proposed bandwidth. When a slave device receives a request from a distant device to initiate a communications flow, the slave device will begin the process illustrated in FIG. 6 to establish the flow of packets to the distant device. Likewise, the distant device will do the same, at its end, to establish a flow of packets towards the local slave device.

- [27] Suppose that a virtual connection is to be established between two nodes on the network to support a telephone voice connection. A voice-over-IP connection may require a 64 kilobits per second transfer rate. This rate could be accomplished by transmitting one 80-byte payload packet every 10 milliseconds. In a like manner, a video stream could also be established. A video stream would typically impose higher bandwidth requirements on the network. For instance, an MPEG-2 encoded video flow could typically require 8 megabytes per second transfer rate. This could be accomplished on an Ethernet LAN, each packet could comprise up to 1,500 bytes of payload. The video could be transmitted using approximately 13 packets, sent every 10 milliseconds.

- [28] Each device repeatedly transmits to the intended recipient according to the agreed delivery schedule. To support a voice-over-IP connection, for example, the transmitter could transmit an 80-byte packet every 10 milliseconds. It could also transmit a 160 byte packet, at 20 millisecond intervals. For a streaming video connection, more packets are required. In this case the transmitter would transmit larger packets, and groups of them, at a more frequent rate.
- [29] While the invention has been described with respect to specific examples including presently preferred modes of carrying out the invention, those skilled in the art will appreciate that there are numerous variations and permutations of the above described systems and techniques that fall within the spirit and scope of the invention as set forth in the appended claims. Any of the method steps described herein can be implemented in computer software and stored on computer-readable medium for execution in a general-purpose or special-purpose computer, and such computer-readable media is included within the scope of the intended invention. The invention extends to not only the method but also to computer nodes programmed to carry out the inventive principles. Numbering associated with process steps in the claims is for convenience only and should not be read to imply any particular ordering or sequence.